BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

Detail 1(1- 1)

Publication No. : 1020020017291 (20020307)
Application No. : 1020000050505 (20000829)
Title of Invention : DEVICE AND METHOD FOR COMPRESSING PROTOCOL HEADER IN COMMUNICATION SYSTEM
Document Code : A
IPC : H04L 29/06
Priority :
Applicant : SAMSUNG ELECTRONICS CO., LTD.
Inventor : LEE, SEONG WON

Abstract :

PURPOSE: A device and a method for compressing protocol header in a communication system are provided to increase efficiency of a communication link by reducing header configuration of UDP/IP/PPP protocols, thereby speeding up the communication link.

CONSTITUTION: A header encapsulator(401) receives UDP (User Datagram Protocol), IP(Internet Protocol), PPP(Point to Point Protocol) packets from an upper protocol layer and extracts headers from the packets for compressing the headers as compression header at the predetermined size and transmitting to a lower protocol layer. A header decapsulator(402) receives the compressed packets through a communication link and releases the compressed packets for disassembling the compressed packets into the original UDP, IP, PPP packets and transmitting the packets to the upper protocol layer. A header database(403) stores information for header compression and release of the encapsulator(401) and the decapsulator(402). A connection tracer(404) drives functions related to connection set-up and connection release of the PPP.

Legal Status :

1. Appliaction for a patent (20000829)

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl. [7]
H04L 29/06

(11) 공개번호 특2002-0017291
(43) 공개일자 2002년03월07일

(21) 출원번호 10-2000-0050505
(22) 출원일자 2000년08월29일

(71) 출원인 삼성전자 주식회사
윤종용
경기 수원시 팔달구 매탄3동 416

(72) 발명자 이성원
경기도성남시분당구서현동91한양아파트327동807호

(74) 대리인 이건주

심사청구 : 없음

(54) 통신시스템에서 프로토콜 헤더 압축장치 및 방법

요약

본 발명에 따른, UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 UDP,IP,PPP 패킷 헤더들을 소정 크기의 압축헤더로 압축하기 위한 방법은, 상기 압축헤더는 L,R,U,M 플래그, 채널식별자 필드(CII-ID), 길이정보 필드(LENGTH), IP정보필드(IP-INFO), 프로토콜 필드(PROTOCOL), 체크섬 필드(UDP CHECKSUM), 부가필드(OPTIONAL FIELD)를 적어도 포함하며, 상위 프로토콜 계층부로부터 상기 UDP,IP,PPP 패킷들을 수신하고, 설정된 통신링크에 대한 연결식별정보를 상기 채널식별자 필드에 기록하고 상기 R플래그를 세팅하는 과정과, 상기 UDP 패킷의 길이 필드 값을 상기 길이정보 필드에 기록하고, 상기 L플래그를 설정하는 과정과, 상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 상기 IP정보필드에 기록하는 과정과, 상기 PPP 패킷의 프로토콜 필드 값을 상기 프로토콜 필드에 기록하는 과정과, 상기 UDP 패킷의 체크섬 필드 값을 상기 체크섬 필드에 기록하고, 상기 U플래그를 세팅하여 기본압축헤더를 생성하는 과정을 포함한다.

대표도
도 4

색인어
RTP, UDP, IP, PPP, Header compression, internet phone, packet voice

명세서

도면의 간단한 설명

도 1은 TCP 패킷 구조를 도시하는 도면.

도 2는 IPv4 패킷 구조를 도시하는 도면.

도 3은 종래기술에 따른 Van Jacobson의 방안에 따라 압축된 TCP/IP 패킷 구조를 도시하는 도면.

도 4는 본 발명에 따른 UDP/IP/PPP 제어정보 압축 및 해제 장치를 도시하는 도면.

도 5는 IPv6 패킷 구조를 도시하는 도면.

도 6은 본 발명에 따른 패킷 헤더 데이터베이스(packet header database) 구조를 도시하는 도면.

도 7은 UDP 패킷 구조를 도시하는 도면.

도 8은 PPP 패킷 구조를 도시하는 도면.

도 9는 본 발명에 따른 UDP/IP/PPP 제어정보 압축에 의해 생성된 패킷 구조를 도시하는 도면.

도 10은 본 발명에 따른 변경필드의 구조를 도시하는 도면.

도 11은 본 발명에 따른 부가필드의 구조를 도시하는 도면.

도 12는 본 발명의 실시 예에 따른 유선환경에서 사용되는 UDP/IP/PPP 제어정보 압축 및 해제 장치를 도시하는 도면.

도 13은 본 발명의 실시 예에 따른 무선환경에서 사용되는 UDP/IP/PPP 제어정보 압축 및 해제 장치를 도시하는 도면.

도 14는 RLP 패킷 구조를 도시하는 도면.

도 15는 RTP 패킷 헤더 구조를 도시하는 도면.

도 16은 본 발명에 따른 RTP/UDP/IP/PPP 헤더 압축에 따라 생성된 패킷 헤더 구조를 도시하는 도면.

도 17은 본 발명에 따른 RTP/UDP/IP/PPP 헤더 압축에 따라 생성된 패킷 헤더 구조를 도시하는 도면.

도 18은 본 발명의 실시 예에 따른 호 설정 과정을 도시하는 도면.

도 19a 및 도 19b는 본 발명의 실시 예에 따른 패킷 헤더 압축 과정을 도시하는 도면.

도 20은 본 발명의 실시 예에 따른 압축된 헤더를 해제하는 과정을 도시하는 도면.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 통신시스템에서 프로토콜 헤더 압축 장치 및 방법에 관한 것으로, 특히 유선 혹은 무선의 저속 인터넷 환경에서 UDP(User Datagram Protocol) 기반 비연결형 통신서비스가 원활하게 지원될수 있도록 UDP/IP(Internet Protocol)/PPP(Point to Point Protocol)의 프로토콜 헤더를 줄이는 장치 및 방법에 관한 것이다.

특히, 본 발명은 멀티미디어 서비스를 인터넷에서 지원하는 경우, 가장 많이 사용되는 RTP(Realtime Transport Protocol)를 함께 고려하여 패킷 음성(Packet voice) 및 패킷 비디오 서비스를 저속 인터넷 환경에서 효과적으로 지원하는 방안을 제안한다. 이를 통하여 저속 유선 통신 환경에서 RTP/UDP/IP/PPP 기반 서비스가 원활하게 지원되며, 아울러 CDMA2000과 같은 무선 통신 환경에서도 인터넷 전화 및 인터넷 비디오 서비스를 유선망과의 호환성을 유지하면서 효율적으로 제공할수 있다. 특히 CDMA2000과 같은 무선 환경에서 인터넷 전화를 위하여 사용되는 경우 무선단에서도 기존의 회선형 전화와 동일하거나 보다 낮은 가격으로 서비스를 지원할수 있으며, 유선 망의 비용도 기존 회선형 음성 전화와 비교하여 크게 줄일수 있는 장점을 제공한다.

현재 비연결형 통신 프로토콜인 UDP/IP 패킷의 헤더를 압축하고 해제하는 방안 혹은 RTP 패킷의 헤더를 압축하고 해제하는 방안은 제안되지 않았으며, 관련 종래 기술로서 연결형 통신 프로토콜인 TCP(Transport Control Protocol)/IP의 헤더를 압축하고 해제하는 방안이 있다. 즉, 기존기술에 있어 UDP/IP/PPP 및 RTP의 헤더를 압축하고 해제하는 방안이 없었다.

유사한 분야로서, 현재 TCP/IP의 헤더를 압축하는 기술은 " Van Jacobson" 에 의하여 제안되어져 있다. 이는 인터넷 관련 표준안 단체인 IETF(Internet Engineering Task Force)에서 RFC1144 " Compressing TCP/IP Header" 로 공표되어져 있다. 상기 Van Jacobson은 저속의 시리얼(Serial)링크에서 TCP/IP 패킷의 전송을 원활히 하기 위하여, TCP/IP의 헤더를 PPP에서 압축하고 해제하는 기술을 제안하고 있다. (V. Jacobson, " Compressing TCP/IP Header for Low-Speed Serial Links" , Fed 1990. http://ieff.org/rfc/rfc1144.txt)

상기 Van Jacobson의 방안에서 TCP/IP 헤더 압축은 송신단 PPP의 압축기(Compressor)에서 수행하고, 해제는 수신단 PPP의 신장기(decompressor)에서 수행한다. 참고로 압축을 수행하지 않은 TCP 패킷의 프레임 구조 및 IP 패킷의 프레임 구조가 도 1과 도 2에 각각 도시되어 있다.

상기 압축기는 IP 패킷을 입력정보로 하고, 모든 입력 패킷을 3가지 타입의 형태로 출력한다. 상기 3가지 타입은 TYPE_IP, UNCOMPRESSED_TCP 혹은 COMPRESSED_TCP이다.

상기 " TYPE_IP" 패킷은 상기 압축기에 입력되는 IP의 형태를 압축기능을 수행하지 않고 그대로 유지하여 출력하는 경우이다. 상기 " UNCOMPRESSED_TCP" 패킷은 다른 필드를 수정하지 않고 단지 IP 패킷의 프로토콜(protocol) 필드를 연결번호(connection no 또는Connect No) 필드로 변경하는 경우이다. 상기 " COMPRESSED_TCP" 패킷은 TCP/IP 헤더 압축을 통하여 헤더 구조를 완전히 새로운 구조로 변경하는 경우이다. 상기 압축을 통한 새로운 TCP/IP 헤더 구조가 도 3에 도시되어 있다.

이하 상기 압축기(Compressor)의 동작을 설명한다.

먼저, TYPE_IP로 전송하는 경우에 대해 살펴보면 다음과 같다.

조건1 : 패킷이 TCP 프로토콜의 패킷이 아니고, 혹은 상기 패킷이 IP 패킷의 조각(fragment)이면 상기 TYPE_IP로 전송한다. 이 경우 조각인 IP 패킷은 첫 번째를 제외하고는 TCP 헤더를 갖지 않으며, 따라서 IP 패킷단을 가지는 패킷에 대한 압축은 지원하지 않는다.

조건2 : TCP 패킷의 SYN, FIN 혹은 RST 플래그가 설정되어 있으면 해당 패킷을 TYPE_IP로 전송한다.

상기한 조건들에 해당하지 않는 패킷은 상기 " COMPRESSED_TCP" 혹은 " UNCOMPRESSED_TCP" 로 전송한다.

한편, 상기 " Van Jacobson" 의 제안 방안은 연결상태(connection state)라는 정보를 송신단에서 관리한다. 상기 연결상태(connection state)는 TCP 혹은 IP 헤더에 포함된 필드 값들을 관리하며, 해당 연결(connection)은 IP 패킷의 소스어드레스(source address), 목적지 어드레스(destination address), 그리고 TCP 패킷의 소스포트(source port)와 목적지포트(destination port)로 식별된다.

조건1 : 만일 도착한 IP 패킷과 동일한 주소 및 포트 번호를 가지는 연결상태(connection state)가 없다면, 송신단에서는 해당 패킷에 대한 연결상태(connection state)를 생성하고 해당 패킷을 상기 " UNCOMPRESSED_TCP" 의 형태로서 전송한다.

조건2 : 도착한 IP 패킷과 동일한 주소 및 포트번호를 가지는 연결상태가 있다면, 상기 압축기는 해당 패킷내의 (Protocol Version, Header Length, Type of Service, DF & MF Flags, Fragment Offset, Time to Live, Protocol, Source Address, Destination Address, Source Port, Destination Port, Data Offset, ACK, SYN, RST, and FIN) 필드들의 값이 연결상태에 저장된 값과 동일한지를 점검한다. 즉, 이전에 전송한 패킷의 정보 값과 동일한지를 점검한다. 만약, 해당 필드들이 다른 값을 가지면 해당 패킷은 상기 " UNCOMPRESSED_TCP" 로 전송한다.

한편, 도착한 IP 패킷이 위에 기술한 조건들에 해당되지 않으면, 압축기는 다음과 같은 동작을 계속 수행한다.

조건1 : TCP의 " URG" 플래그가 설정되어 있으면, " urgent data" 필드가 인코딩되고, 도 3의 " change mask" 내의 U 비트가 설정된다. 만약, " URG" 플래그가 설정되어 있지 않으면, " urgent data" 필드의 내용은 이전 패킷과 비교되며, 이 때 내용이 다른 경우 상기 " UNCOMPRESSED_TCP" 패킷을 전송한다.

조건2 : 만약, TCP의 " window" 필드 값이 이전 패킷의 " window" 값과 다른 경우 도 3의 " change mask" 내의 W 비트를 설정하고 상기 " windoe" 필드의 차이 값을 인코딩한다.

조건3 : 만약, TCP의 " ACK" 필드 값이 이전 패킷의 값과 다르면, 도 3의 " change mask" 내의 A 비트를 설정하고, 상기 " ACK" 필드의 차이 값을 인코딩한다. 이 경우, 상기 차이 값이 " 0" 이하 혹은 216-1 이상이면 상기 " UNCOMPRESSED_TCP" 패킷을 전송한다.

조건4 : 만약 TCP의 " sequence number" 필드 값이 이전 패킷의 값과 다르면, 도 3의 " change mask" 내의 S비트를 설정하고, 상기 " sequence number" 필드의 차이 값을 인코딩한다. 이 경우, 상기 차이 값이 " 0" 이하 혹은 216-1 이상이면 상기 " UNCOMPRESSED_TCP" 패킷을 전송한다.

상기와 같이 변경마스크(Change mask)의 U, W, A 및 S비트가 설정되면 다음과 같은 특별 케이스에 대한 작업을 수행한다.

조건1: 만약, 상기 S비트만 설정되면, TCP/IP의 헤더를 제외한 사용자데이터(user data)의 크기를 이전 패킷과 비교한다. 이 경우, 패킷의 크기가 동일하면 상기 변경마스크를 SAWU(" unidirectional data transfer" 에 대한 특별 케이스)로 설정하고, 인코딩된 시퀀스번호(sequence number)를 제거한다. 이 때에 수신 단 해제기(decompressor)가 이전 패킷에 대한 전체 길이 및 헤더 길이를 알고 있으므로 통신에는 문제가 없다.

조건2 : 만약 상기 S와 상기 A비트가 설정되고, 상기 S와 A에 대한 필드 값의 변경 정도가 동일하며 그 변경정도가 다지막 패킷 안의 사용자 데이터의 량이라면, SWU(" echoed interactive" 트래픽에 대한 특별 케이스)를 설정하고 인코딩된 변경사항을 제거한다.

조건3 : 만약 변경된 필드가 없으면서, 사용자 데이터가 없거나 이전 패킷의 데이터를 가지고 있으면 재전송으로 고려하여 상기 " UNCOMPRESSED_TCP" 를 전송한다.

다지막으로 TCP/IP 헤더는 압축된 헤더로 변경되어 전송된다. 이에 해당하는 작업들은 다음과 같다.

먼저, IP 패킷의 Packet ID(Identification) 값에 대한 이전 패킷과의 변경 정도를 계산한다. 만약, 변경 정도가 1이 아니면, change mask의 I 비트를 설정하고 차이 값을 인코딩한다. (Identification은 일반적으로 매 패킷의 전송 후에 1씩 증가함) 그리고, 만약 TCP의 PUSH가 설정되어 있으며, change mask의 P 비트를 설정한다. 이후, TCP/IP의 헤더 정보를 송신단의 연결상태에 저장한다. 그리고, 새롭게 생성된 TCP/IP 헤더를 패킷에 삽입한다. 새롭게 생성되는 패킷은 다음과 같은 정보를 포함한다.

① 인코드eld된 변경 값들

② 도착한 TCP 헤더에서 추출한 checksum 필드 값

③ 연결번호 (connection number) – 이전 패킷과 현재의 패킷이 다른 경우만 삽입

④ Change mask

수신단의 해제기(decompressor)는 상기한 압축기(compressor)의 동작과 비교해 볼때 보다 단순한 처리를 수행한다. 상기 해제기는 (패킷 길이, 패킷 타입, 연결상태 정보)를 입력 파라메타로 동작한다. 상기 해제기는 4가지 형태의 패킷을 수신할 수 있다. 이들은 송신 단이 생성한 " TYPE_IP" , " COMPRESSED_TCP" , " UNCOMPRESSED_TCP" 와 전송상의 에러로 발생하는 " TYPE_ERROR" 가 된다.

조건1 : 만약 TYPE_ERROR 혹은 식별 불가능한 패킷이 수신되면, 수신단은 'toss' 플래그를 설정한다. 아울러, 해당 패킷과 이후에 도착하는 패킷들을 C비트가 설정된 COMPRESSED_TCP 혹은 UNCOMPRESSED_TCP가 도착 한 때까지 폐기한다.

조건2 : 만약 TYPE_IP 패킷이 도착하면, 수정되지 않은 복사본이 리턴되며, 관련 상태 정보는 수정되지 않는다.

조건3 : 만약 UNCOMPRESSED_TCP 패킷이 도착하면, IP 프로토콜 필드와 수신단의 상태 정보를 점검한다. 이 경우, 무효(invalid)한 패킷인 경우는 'toss' 플래그를 설정하고 해당 패킷을 폐기한다. 유효(Valid)한 패킷인 경우에는 " toss" 플래그를 해제하고, 해당 패킷의 정보를 수신단의 상태 정보로 저장한다. 그리고, 수신된 정보를 기반으로 하여 압축 해제된 패킷 헤더를 생성한다.

상기 기술한 조건들에 해당하지 않는 패킷은 " COMPRESSED_TCP" 패킷이다. 따라서, 수신된 패킷 헤더의 압축 해제를 위해서는 연결번호를 통하여, 수신단의 상태 정보로부터 이전에 수신된 패킷의 정보를 추출한다. 그리고, 수신된 패킷의 정보와 이전 패킷의 정보를 기반으로 하여 새롭게 생신된 헤더를 만들어야 한다. 이에 해당하는 내용은 다음과 같다.

조건1 : 만약 C비트가 설정되어 있으면 해당 패킷의 연결번호(connection number)를 통하여 수신 단의 상태 정보를 검색한다. 만약, 해당 연결번호에 대한 정보가 수신단에 없는 경우는 해당 패킷을 폐기하고, " toss" 플래그를 설정한다. 상기 C비트가 설정되어 있지 않은 경우에는 마지막으로 이전에 수신된 패킷의 연결번호를 해당 패킷의 연결번호로 사용하며, " toss" 플래그를 해제한다.

조건2 : 만약 상기 C 비트가 설정되어 있지 않고, " toss" 플래그가 설정된 경우에는 해당 패킷을 폐기한다.

아울러, 세부적인 필드 재구성 절차는 다음과 같다.

먼저, TCP checksum 필드는 수신된 값 그대로 재구성 버퍼에 저장한다.

여기서, 만약 P 비트가 설정되어 있으면, TCP PUSH 비트를 설정하여 재구성 버퍼에 저장한다. 그렇지 않으면, 설정하

지 않은 상태로 저장한다. 만약 S, A, W 및 U 비트가 모두 설정되어 있으면 (unidirectional data), 이전의 마지막에 수신한 패킷의 사용자 데이터 필드 크기에서 수신한 패킷의 TCP/IP 헤더 길이를 뺀 값을 계산한다. 그리고, 계산된 값을 TCP 시퀀스번호(sequence number)에 더한 후 재구성 버퍼로 저장한다. 만약 S, W 및 U 비트가 설정되어 있으면 (terminal traffic), 이전의 마지막에 수신한 패킷의 사용자 데이터 필드 크기를 TCP 시퀀스 번호와 ack 필드에 더한 후 재구성 버퍼에 저장한다. 마지막으로, 위의 경우에 해당하지 않는 경우는 다음의 절차를 수행한다.

만약 U 비트가 설정되어 있으면, TCP URG 비트가 설정되어 재구성 버퍼로 저장되며, urgent pntr 필드의 값이 TCP Urgent Pointer 필드 값으로 할당되어 재구성 버퍼로 저장된다. 만약 W 비트가 설정되어 있으면, _d window 필드의 값을 이전에 수신한 TCP window 필드 값에 더하여 재구성 버퍼로 저장한다. 만약 A 비트가 설정되어 있으면, _d ack 필드의 값을 이전에 수신한 TCP ack 필드 값에 더하여 재구성 버퍼로 저장한다. 만약 S 비트가 설정되어 있으면, _d sequence 필드의 값을 이전에 수신한 TCP sequence 필드 값에 더하여 재구성 버퍼로 저장한다. 만약 I 비트가 설정되어 있으면, _d IP ID 필드의 값을 이전에 수신한 TCP Identifier 필드 값에 더하여 재구성 버퍼로 저장한다. 이후, 헤더 및 데이터 정보의 길이를 계산하여 재구성 버퍼로 저장하고, 또한 IP의 checksum 값을 재계산하여 재구성 버퍼로 저장한다. 이를 통하여 재구성된 TCP/IP 패킷이 완성된다.

전술한 바와 같이, 상기 " Van Jacobson" 이 제안한 방안의 문제점은 TCP/IP를 대상으로 고려하였기에 본 발명이 고려하고자 하는 UDP/IP의 환경에는 적합하지 않다는 점과 함께, TCP/IP의 헤더 정보중 이전 패킷과의 '차이 값'으로 전송하지 않는 필드의 정보(일반적으로 변경이 거의 없는 필드의 정보)가 변경되는 경우에는 패킷 헤더 전체를 전송해야 하는 문제점이 있다. 즉, 변경된 필드의 정보만을 전송하는 기능이 없다.

아울러, 헤더의 최소 크기 구조를 가정하여 기법을 제안하므로서, TCP/IP 헤더에 부가(Options) 정보가 실리는 경우에는 별다른 지원 방안이 기술되어 있지 않다. 따라서, 이러한 경우에는 전체 헤더를 그대로 전송해야하는, 즉 압축 방안이 지원될 수 없는 상황에 직면한다.

또한, 상기 " Van Jacobson" 의 제안방안은 '차이 값'을 전송하는 방안에 전적으로 의지하므로서, '차이 값'을 전송하던 헤더가 중간에 에러가 발생하여 폐기되는 경우에는 연속적으로 수신되는 패킷들을 전체 헤더가 압축되지 않은 패킷이 수신될 때 까지 폐기하고, TCP의 재전송 기능을 통하여 에러를 복구하는 기능을 지원해야 한다. 그러나, 상기 " Van Jacoban" 방안은 재전송을 고려하지 않으며, 주로 지연에 민감한 트래픽을 전송하는 UDP 패킷에서는 치명적인 성능 저하를 야기하는 문제점을 가진다. 즉, 인터넷 전화 같은 경우 에러가 발생하는 경우에서는 전체 헤더가 수신되는 경우가 발생할 때까지 패킷들이 폐기되므로 통화 불능 상태에 빠지는 결과를 초래한다.

발명이 이루고자 하는 기술적 과제

따라서 본 발명의 목적은 저속 유무선 직렬(serial) 통신 장치들에 있어서 UDP, IP, IPv4 & IPv6 및 PPP 프로토콜을 기반으로 하는 통신 서비스가 효과적으로 지원되는 방안을 제공함에 있다.

본 발명의 다른 목적은 현재 인터넷 멀티미디어 서비스의 주요 기술로서 사용되고 있는 UDP, IP, IPv4 & IPv6 및 PPP 프로토콜의 헤더 크기를 최소화 함으로서, 보다 많은 통신 용량을 실제 사용자 트래픽의 송수신에 활용할 수 있는 장치 및 방법을 제공함에 있다.

본 발명의 또 다른 목적은 무선 통신망에서 기존 회선형 음성 서비스가 사용하는 대역 요구 수준 이하의 대역에서 효과적으로 인터넷 전화를 사용하게 할 수 있도록 하는 방안을 제안함에 있다.

본 발명의 또 다른 목적은 비연결형 UDP 통신 프로토콜에 기반하는 실시간 멀티미디어 서비스들을 지원하기 위한 장치 및 방법을 제공함에 있다.

상기 목적을 달성하기 위한, UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 UDP,IP,PPP 패킷 헤더들을 소정 크기의 압축헤더로 압축하기 위한 방법이, 상기 압축헤더는 L,R,U,M 플래그, 채널식별자 필드(CH-ID), 길이정보 필드(LENGTH), IP정보필드(IP-INFO), 프로토콜 필드(PROTOCOL), 체크섬 필드(UDP CHECKSUM), 부가필드(OPTIONAL FIELD)를 적어도 포함하며, 상위 프로토콜 계층부로부터 상기 UDP,IP,PPP 패킷들을 수신하고, 설정된 통신링크에 대한 연결식별정보를 상기 채널식별자 필드에 기록하고 상기 R플래그를 세팅하는 과정과, 상기 UDP 패킷의 길이 필드 값을 상기 길이정보 필드에 기록하고, 상기 L플래그를 설정하는 과정과, 상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 상기 IP정보필드에 기록하는 과정과, 상기 PPP 패킷의 프로토콜 필드 값을 상기 프로토콜 필드에 기록하는 과정과, 상기 UDP 패킷의 체크섬 필드 값을 상기 체크섬 필드에 기록하고, 상기 U플래그를 세팅하여 기본압축헤더를 생성하는 과정을 포함하는 것을 특징으로 한다.

발명의 구성 및 작용

이하 본 발명의 바람직한 실시 예를 첨부된 도면의 참조와 함께 상세히 설명한다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 동일 부호를 가지도록 하였다. 또한 본 발명을 설명함에 있어서, 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략한다.

본 발명이 가장 효과적으로 사용될 수 있는 분야는 저속의 유선망과 함께, 기존 무선 통신망인 IS95A, IS95B, cdma2000 및 IMT2000이다. 상기한 무선 통신망들은 유선망과 비교하여 저속의 통신 채널을 가입자에게 지원한다. 따라서, 제한된 무선 채널을 효과적으로 사용하기 위해서는 패킷 헤더와 같은 제어 정보의 최소화가 요구된다.

특히, 최근 폭발적으로 확성화되는 인터넷 전화의 경우에도 현재의 무선 통신망 구조 및 통신 프로토콜로서는 무선 통신망에서 지원이 불가능하거나 비용이 매우 비싸 시장성이 없지만, 본 발명에서 제안하는 방안을 사용하면 기존 회선형 전화에 비교하여 낮은 가격의 구현이 가능해진다.

현재 급격한 성장을 보이고 있는 인터넷 전화는 기본적으로 이동 전화와 같이 음성을 압축하는 방식을 취한다. 이를 '음성 압축'이라 하며 이는 보코더(vocoder)라 하는 압축 장치를 통하여 이루어진다. 보코더의 핵심은 음성 압축 방식에 있으며 이를 수행하는 모듈을 코덱(codec)이라 한다.

상기 인터넷 전화를 위하여 사용하는 코덱은 크게 ITU(International Telecommunication Unions)-T의 표준안인 G.729와 G.723.1이 있다. 상기 ITU-T G.729는 8kbps의 대역을 요구하며, 10ms 마다 음성 프레임을 생성한다. 이때 음성이 활성화된 (talk) 구간에서는 80bits를 생성하고, 비활성화된 묵음 (silence) 구간에서는 15bits의 음성 트래픽을 생성한다. 또한 유선 망에서 활용시에는 비활성화 구간의 트래픽을 제거하는 'silence compression' 기능을 통하여 통신 대역을 보다 효과적으로 사용하도록 유도한다. 따라서, G.729 기반의 인터넷 전화는 활성화시에 매 10ms마다 264bits의 헤더 정보와 80bits의 음성 정보를 생성한다.

한편, 상기 ITU-T G.723.1은 6.3kbps 혹은 5.4kbps의 대역을 요구하며, 30ms 마다 음성 프레임을 생성한다. 이때 음성 트래픽은 활성화시에 각각 192bits 혹은 160bits의 트래픽을 생성한다. G.723.1도 G.729와 마찬가지로 'silence compression' 기능을 제공하며, 무선 환경에 적합하도록 지원하는 방안을 별도로 제공한다. 따라서, G.723.1을 사용하는 경우에는 매 30ms마다 264bits의 헤더 정보와 192bits 혹은 160bits의 음성 트래픽 정보가 생성된다.

이와 관련하여, 기존 IS95A, IS95B, PCS(Personal Communication Service) 및 GSM(Global System for Mobile communications)과 같은 무선 통신 시스템에서는 9.6kbps 혹은 14.4kbps의 저속 채널을 통하여 음성 서비스를 지원해 왔다. 따라서, G.729 혹은 G.723.1에 기반하는 인터넷 전화 서비스의 지원을 위하여 무선 대역이 9.6kbps 혹은 14.4kbps를 넘어서는 경우에는 기본적으로 무선단에서의 비용이 더욱 증가하는 결과를 초래한다. 따라서, G.729 혹은

G.723.1을 지원하는 인터넷 전화의 경우도 무선 상황에서 동일하게 9.6kbps 혹은 14.4kbps의 대역에서 지원되거나 그 이하의 속도에서 지원되어야 한다.

그러나, 앞서 기술하였듯이 UDP, IPv4 & IPv6, PPP의 헤더 압축 기능이 없을 경우, G.729 및 G.723.1은 각각 688 bits/20ms, 456bits(혹은 424bits)/30ms의 용량을 가지며, 이는 cdma2000을 고려하여 9.6kbps의 이동 통신망이 지원할 수 있는 160bits/20ms 혹은 14.4kbps의 이동 통신망이 지원할 수 있는 256bits/20ms를 훨씬 상회하는 요구치이다. 특히, 이 경우 인터넷 전화에서 점차 사용이 확산되고 있는 RTP의 헤더를 고려하면 상회하는 요구치는 더욱 커진다. 아울러, IPv6의 경우는 어드레스 필드의 크기가 IPv4의 4바이트에서 16바이트로 증가함에 따라 더욱 오버헤드를 필요로 한다. 결론적으로 현재의 UDP, IPv4 & IPv6, PPP 및 RTP의 헤더를 그대로 사용하는 경우에는 효과적인 가격의 UDP 기반 서비스 및 인터넷 전화가 무선 통신 시스템에서 지원될 수 없다. 아울러, 고가의 제속 무선 대역을 대부분 헤더 전송에 사용하므로서 효율이 매우 낮아지는 결과를 초래한다.

따라서, 본 발명은 무선 통신망에서도 기존 회선형 음성 서비스가 사용하는 대역 요구 수준 이하의 대역에서 효과적으로 현재의 인터넷 전화를 사용하게 할 수 있는 방안을 제안한다. 특히, 이러한 방안은 기존의 회선형 전용 라인을 중심으로 하는 무선 통신망 구조를 패킷 통신 방식으로 재편하도록 유도하므로서 무선 망의 구축에 필요한 비용을 현격히 줄일 수 있다.

이하 도면의 참조와 함께 상세히 설명한다.

본 발명의 실시예에 따른 UDP/IPv4& IPv6/PPP 헤더 압축 및 해제 장치인 헤더 옵티마이저(Header Optimizer : 이하 옵티마이저라 칭함)는 도 4에 도시된 바와 같이, 크게 헤더 인캡슐레이터(Header Encapsulator : 이하 인캡슐레이터라 칭함)401, 헤더 디캡슐레이터(Header Decapsulator : 이하 디캡슐레이터라 칭함)402, 패킷 헤더 데이터베이스(Packet Header Database : 이하 헤더 데이터베이스라 칭함)403 및 연결추적기(connection tracer)404로 구성된다. 이하 설명에서 IP란 IPv4와 IPv6를 통칭하는 용어이다.

본 발명에서 제안하는 상기 옵티마이저의 구현 형태는 다음과 같은 방식으로 이루어 질 수 있다. 첫 번째, Van Jacobson의 TCP/IP 헤더 압축 기법처럼, PPP 링크 프로토콜의 확장된 기능으로 구현할 수 있고, 두 번째 기존의 PPP와 하위 통신 프로토콜의 사이에 신규의 프로토콜 계층으로서 구현할수 있으며, 세 번째 무선 통신망에서 RLP(Radio Link Protocol) 통신 프로토콜의 확장된 기능으로서 구현할수 있다. 본 발명은 상기한 형태 중에서 어떠한 방식을 취하든지 간에 성능의 차이는 없다. 발명의 설명을 용이하게 하기 위하여 별도의 구분이 없는 한 첫 번째 방식에 의하여 본 발명을 적용하는 것을 가정하여 설명하도록 한다. 또한, 본 발명은 단방향(simplex)과 양방향(half duplex, full duplex) 통신 방식을 모두 지원한다. 따라서, 상기 옵티마이저는 헤더 압축을 위한 인캡슐레이터와 디캡슐레이터의 쌍(pair)을 가진다.

상기 도 4를 참조하면, 상기 인캡슐레이터401은 PPP 프레임(UDP/IP/PPP PACKET)을 입력으로 하여, UDP/IPv4& IPv6/PPP 헤더의 압축 기능을 수행한 후 출력 통신 링크 혹은 하부의 통신 계층에 전달하는 기능을 수행한다. 상기 디캡슐레이터402는 통신 링크를 통하여 수신된 PPP 프레임에서 압축된 UDP/IPv4& IPv6/PPP 헤더를 추출하여 원래의 형태로 압축을 해제한 후, 프로토콜 구조상의 상위 프로토콜로 전달하는 기능을 수행한다. 헤더 데이터베이스(H-DB)403은 UDP/IPv4& IPv6/PPP 헤더의 압축을 수행하는데 필요한 정보를 저장하는 메모리이다. 상기 연결추적기(Connection Tracer)404는 PPP의 연결 설정 및 해제 관련 메세지를 분석하거나 PPP의 연결설정 및 해제기능과 연계하여 PPP 연결의 신규 설정 및 해제를 검출하고 관련된 옵티마이저의 기능을 구동하는 기능을 수행한다.

상기 헤더 데이터베이스403은 첨부된 도면 도 6에 도시된 바와 같이, 압축을 수행할 논리적 통신 연결을 식별할 연결 식별자(CONN-ID : Connection Identifier)와 해당 연결과 관련된 정보 저장 부분으로서 구성된다. 상기 연결식별자는 구현 목적에 따라 다음의 정보를 할당할수 있다.

경우1 : < Source Address(IP), Destination Address(IP), Source Port(UDP), Destination Port(UDP)>

경우2 : < Source Address(IP), Destination Address(IP), Source Port(UDP), Destination Port(UDP), Protocol Field(IP)>

경우3 : < Source Address(IP), Destination Address(IP), Source Port(UDP), Destination Port(UDP), Protocol Field(IP), Protocol Field(UDP)>

상기 경우1은 일반적인 용도로 사용하는 식별자이다. 즉, 송신단과 수신단의 IP 주소 및 UDP 포트 번호가 다른 경우 다른 연결식별자(CONN-ID)를 할당한다. 경우2와 경우3은 상기 경우1에 추가적으로 IP 혹은 UDP의 프로토콜 필드 값을 활용하는 경우로서, 보다 구체적으로 상기 연결식별자(CONN-ID)를 구분하고자 하는 경우에 활용한다. 특히 IPv6를 사용하는 경우 IPv6의 " Traffic Class" 및 " Flow Label" 을 이용한 식별도 고려할수 있다. 상기 연결식별자(CONN-ID)로 식별되는 각 링크에 대한 정보는 도 5에 도시된 UDP/IPv4& IPv6/PPP의 헤더를 구성하는 각 필드들로서 정의된다. 따라서, 상기 헤더 옵티마이저는 상기 연결식별자(CONN-ID)로 구분되는 연결들의 UDP/IPv4& IPv6/PPP 헤더 정보를 계속 상기 헤더 데이터베이스403에 갱신한다. 이를 통하여, 상기 인캡슐레이터401은 전송해야할 필드와 값을 계산할 수 있으며, 수신단에서는 패킷 헤더의 재구성을 위한 정보를 검색할 수 있다.

이하 본 발명에 따른 구체적인 동작을 살펴본다.

본 발명의 동작은 크게 4가지 단계로 구성될 수 있다. 가장 먼저 이루어지는 단계는 '호 설정 과정'으로서, PPP 종단간 연결을 설정하는 단계이다. 이 때에 수행되는 주요한 작업은 해당 연결에 대한 연결식별자(CONN-ID)의 할당과 상기 헤더 데이터베이스에 상기 연결식별자에 관련된 필드들을 생성하고 초기화하는 일이다. 상기 '호 설정 과정'을 통하여 PPP 종단간에 연결이 이루어지면, 상호간에 PPP 트래픽을 송수신할 수 있는 단계에 진입한다. 상기 단계에서는 상기 인캡슐레이터401가 상위 프로토콜로부터 수신되는 UDP/IPv4& IPv6/PPP 패킷의 헤더를 압축하고 관련 헤더 데이터베이스(H-DB) 정보를 갱신하는 '패킷 압축 과정' 관련 작업이 이루어진다. 아울러, 디캡슐레이터402에서는 통신 링크 혹은 하위 통신 프로토콜로부터 수신된 PPP 프레임을 수신하여, 해당 프레임내의 UDP/IPv4& IPv6/PPP 헤더의 압축을 해제하여 원래의 UDP/IPv4& IPv6/PPP 헤더 형태로 복원하는 '패킷 압축 해제' 관련 작업이 이루어진다. 그리고 마지막으로 '호 해제 과정' 단계로 구성된다. 이하 각각의 과정에 대한 상세한 절차를 설명한다.

< 호 설정 과정>

도 18은 본 발명의 실시 예에 따른 호 설정 과정을 도시하고 있다.

상기 도 18을 참조하면, 먼저, 1801단계에서 사용자(혹은 상위 프로세스)의 요청에 의하여 종단간에 통신 링크가 설정되면, 상기 옵티마이저는 1803단계에서 해당 연결의 IP 소스어드레스(Source Address), IP 목적지 어드레스(Destination Address), UDP 소스포트(Source Port), UDP 목적지 포트(Destination Port)를 식별하여, 해당 연결에 대한 필드가 상기 헤더 데이터베이스403에 있는지를 검사한다. 여기서, 추가적으로 IP Protocol Field(IP 프로토콜 필드) 및 UDP Protocol Field(UDP 프로토콜 필드)를 사용할 수도 있다. 이때, UDP 패킷이 아닌 경우는 옵티마이저가 관여하지 않는다. 이 경우 앞서 설명한 바와 같이, IPv4 헤더의 " Traffic Class" 및 " Flow Label" 도 식별 정보로 고려할 수 있다.

상기 헤더 데이터베이스403에 대한 검사 후에 1805단계에서 동일한 정보를 가지는 연결식별자가 존재한다고 판단되면 1815단계로 상기 통신링크가 설정된 종단간에 하나 이상의 복수링크가 허용되는지를 검사한다. 만약 같은 종단간에 하나 이상의 복수 링크가 허용되면 1817단계로 진행하여 현재 상기 헤더 데이터베이스403에 존재하는 연결식별자를 동일하게 사용하여, 해당 연결식별자에 관련된 필드들의 값을 공유하는 방식을 지원하도록 한다. 반면 그렇지 않은 경우, 1807단계로 진행하여 해당 종단간의 연결식별자를 할당함에 있어 상기한 '경우1 → 경우2 → 경우3'의 방식으로 확장함으로서 두 연결간의 식별을 지원하도록 한다. 그리고 상기와 같이 해당 연결에 대한 연결식별자(CONN-ID)가 설정되면 상기 헤더 데이터베이스403에 상기 해당 연결식별자에 대한 필드들을 생성한다.

이후, 1811단계에서 해당 연결을 통하여 소정 전송 시점에서 상위 프로토콜 계층으로부터 첫 번째 패킷(UDP,IP,PPP 패킷들)이 수신되는지를 검사한다. 만일, 상기 첫 번째 프레임이 수신되면 1813단계에서 수신된 첫 번째 프레임의 UDP/IP/PPP 헤더의 정보를 해당 연결의 연결식별자에 대응하는 상기 헤더 데이터베이스403의 필드들에 복사한다. 상기 복사가 종료되면, 첫 번째 패킷의 가장 첫 번째 필드로서 연결식별자의 값이 추가되고, 이후의 원래 UDP/IP/PPP 헤더 필드들은 압축 과정 없이 수신단으로 전송한다.

송신단의 인캡슐레이터401이 첫번째 패킷에서 상기 헤더 데이터베이스403으로 복사하는 필드들은 다음과 같다. 일부 필드들의 길이는 조건에 따라 달라질 수 있으며, 아래 기술한 길이는 가장 일반적인 경우를 가정한 경우이다. 여기서 상기 헤더 데이터베이스403에 복사되는 UDP 패킷 필드는 도 7에 도시되고 있고, PPP 패킷 필드는 도 8에 도시되어 있다.

상기 도 7에 도시된 바와 같이, 상기 헤더 데이터베이스403에 복사되는 UDP 패킷 필드는 8 bytes (64 bits)로 구성되며, 전송 프로세스의 포드 번호가 기록되는 소스포드(Source Port) 필드(16 bits), 수신 프로세서 컨텍스트의 포드 번호가 기록되는 목적지 포트(Destination Port) 필드(16 bits), 데이터 그램의 옥텟 단위 크기(헤더 포함)가 기록되는 길이(Length) 필드(16 bits) 및 에러 검출자 정보가 기록되는 체크섬(Checksum) 필드(16 bits)필드로 구성된다.

그리고, 상기 헤더 데이터베이스403에 복사되는 IPv4 패킷 필드는 20 bytes (160 bits) + 옵션(option) 필드 길이로 구성되며, 인터넷 헤더의 포맷을 기록하는 Version 필드(4 bits), 인터넷 헤더의 길이를 기록하는 IHL 필드(4 bits), 서비스 종류 및 구분자를 기록하는 Type of Service 필드(8 bits), 헤더와 사용자 데이터부분을 더한 길이를 기록하는 Total Length 필드 (16 bits), IP 패킷 식별자(데이타 그램의 segment시 이용)를 기록하는 Identification 필드(16 bits), Fragment 제어 플래그를 기록하는 Flags 필드(3 bits), Fragment시 해당 정보의 데이터 그램내 위치 표시자를 기록하는 Fragment Offset 필드(15 bits), 망에서의 패킷 체류 허용시간을 기록하는 Time to Live 필드(8 bits), 인터넷 데이터 그램의 사용 프로토콜 식별자를 기록하는 Protocol필드(8 bits), 헤더 에러 검출자를 기록하는 Header Checksum 필드(16 bits), 송신자 주소를 기록하는 Source Addr 필드(32 bits) 및 수신자 주소를 기록하는 Destination Addr필드(32 bits)로 구성된다.

그리고, 상기 헤더 데이터베이스403에 복사되는 IPv6 패킷 필드는 40 bytes (320 bit)로 구성되며, 인터넷 헤더의 포맷을 기록하는 Version 필드(4 bit), 서로 다른 클래스와 우선순위를 가지는 패킷 식별를 기록하는 Traffic Class 필드(8bit), 송신단과 수신단의 패킷 스트림 식별을 기록하는 Flow Label 필드(20 bit), 패킷이 포함하고 있는 정보량을 기록하는 payload length 필드(16bit), 헤더정보를 기록하는 Next Header 필드(8bit), 패킷망에서의 최대 허용 라우팅 홉을 제어하는 Hop Limit 필드(8bit), 송신단 IPv6 주소를 기록하는 Source Addr(128 bit) 및 수신단 IPv6 주소를 기록하는 Destination Addr 필드(128 bit)로 구성된다.

마지막으로, 상기 도 8에 도시된 바와 같이, 상기 헤더 데이터베이스403에 복사되는 PPP 패킷 필드는 5/7 bytes(10/56 bits)이며, '11111111'가 기록되는 Address필드(8 bits), '00000011'가 기록되는 Control필드(8 bits), 데이터 그램을 이용하는 프로토콜 식별자가 기록되는 Protocol 필드(8/16 bits) 및 에러 점검자가 기록되는 FCS필드 (16/32 bits)로 구성된다.

상기한 필드들의 정보와 함께, 압축된 헤더의 순서 관리를 위한 시퀀스(Sequence) 필드를 상기 헤더 데이터베이스403에 생성한다. 상기 시퀀스 필드는 IP시퀀스를 65로 나눈 나머지 값을 저장한다.

한편, 수신단의 디캡슐레이터402는 첫 번째 패킷을 수신하면, 상기 인캡슐레이터401과 마찬가지로 헤더 데이터베이스403을 초기화하여 수신한 프레임의 첫번째 헤더 필드인 연결식별자를 식별자로 하여 설정된 연결의 정보를 포함한 필드를 생성한다. 그리고, 시퀀스 필드 값을 자신의 헤더 데이터베이스403에 저장한 후, 상기 디캡슐레이터401은 수신된 UDP/IPv4& IPv6/PPP 헤더의 정보를 상기 헤더 데이터베이스403로 그대로 복사하고, 해당 패킷의 연결식별자를 제거한후 상위 프로토콜 혹은 사용자에게 전달한다.

< 패킷 헤더 압축 과정>

도 19a 및 도 19b는 본 발명의 실시 예에 따른 패킷 헤더 압축 과정을 도시하고 있다. 본 발명에서 지원하는 UDP/IP/PPP 패킷 헤더 압축 기법은 첫째로 기본적으로 연결 설정 이후에 거의 변경되지 않는 필드들을 '변경이 발생하는 시점'에서만 전송하도록 하는 방안과 함께, 둘째로 수시로 변하는 값의 경우도 쌍방간에 통신이 이루어지는데 필요한 최소한 정보만을 기본으로 전송하고, 필요시에만 전체 필드 값을 보내는 방법을 기본으로 하여 이루어진다. 셋째로 IP의 Options에 해당하는 부가 제어 정보 전송시에도 전체 헤더를 보내던 Van Jacobson의 방안과 달리 추가되는 필드들만을 압축된 헤더에 덧붙여서 전송하는 기법을 제공한다. 특히, 무선 통신에서의 지원도 함께 고려하여 복잡한 재전송을 야기할 수 있는 '차이 값' 전송 방식은 고려하지 않는다. 따라서, '차이 값'만이 전송되다가 에러가 발생하는 경우, 이후에 전체 헤더가 수신될 때까지 헤더를 번역할 수 없어서 패킷들이 모두 폐기되는 Van Jacobson의 문제점을 해결한다.

상기 도 19를 참조하면, 1901단계에서 해당 연결의 연결식별자(CONN-ID) 값을 헤더데이터베이스의 채널식별자(CH-ID) 필드로 복사하고, R필드를 " 0" 으로 설정한다. 그리고, 1903단계에서 UDP 패킷의 길이가 " 255" 옥텟(octet)을 초과하는지 검사한다. 만일, 상기 패킷 길이가 255 옥텟을 초과하지 않은 경우 1905단계로 진행하여 UDP 헤더의 길이필드를 압축된 헤더의 길이필드로 저장하고 L 플래그를 " 0" 으로 설정한다. 반면, 길이가 255 옥텟을 초과하는 경우 1933단계로 진행하여 L 플래그를 " 1" 로 설정하고, 16비트의 길이 필드를 사용한다.

그리고, 1907단계에서 IPv4를 사용하는지를 검사한다. 만일, 사용하는 경우 1931단계로 진행하여 IPv4 패킷의 식별자(Identification) 필드 값을 하기 < 수학식 1> 과 같이 수정하여 수정된 값을 'IP-INFO' 필드에 저장한다.

수학식 1
IP-INFO = IPv4 Identification % 65

만일, IPv6를 사용하는 경우 1909단계로 진행하여 상기 'IP-INFO" 필드를 IPv6 헤더의 Next Header정보로서 삽입한다.이후, 1911단계에서 PPP 헤더의 Protocol 필드 값을 Protocol 필드로 저장한다. 여기서, RFC1661의 프로토콜 헤더 압축(Protocol Header Compression)에 따라 PPP는 8 bits 필드로 고정되도록 한다. 그리고, 1913다계에서 시비스가 UDP 체크섬(checksum)이 요구되는지 검사한다. 만일, 체크섬이 요구되면 UDP 헤더의 Checksum 필드 값을 UDP Checksum 필드에 저장하고, U 플래그를 1로 설정한다. 만약, 트래픽 속성이 체크섬을 필요로 하지 않으면, U 플래그를 " 0" 으로 하고 UDP checksum 필드를 제거한다.

상기와 같은 과정을 통해 생성된 헤더는 UDP/IP/PPP 기반 통신시에 항상 유지되어야 하는 필드들이다. 상기한 과정에서 제외된 UDP/IP/PPP 프로토콜의 필드들은 일반적으로 패킷 송수신 과정에서 변하지 않는 필드들을 의미한다. 따라서, 일반적으로 고정된 값을 가지는 필드들이 변경되는 경우에는 해당 필드의 값을 상기와 같이 생성된 기본 헤더 구조에 부가하는 형태를 따르게 된다. 첨부된 도면 도 9는 상기와 같이 생성된 기본 헤더 구조를 도시하고 있다.

고정된 값을 가지는 필드들이 변경되는 경우 해당 필드의 값을 기본 헤더 구조에 부가하기 위해서는 하기 < 표 1> 과 같은 필드 식별 테이블이 필요하다.

상기 필드 식별 테이블에서 'NOTE#1'으로 설정된 Total Length, Time to Live, Header Checksum 및 Frame Check Sequence는 패킷마다 다른 값들이 전송되는 필드들이지만, 이들 필드들의 값이 변하더라도 부가 정보로 전송하지는 않는다. 그럼에도 불구하고 테이블에 해당 필드들을 정의한 이유는, 향후 제어 목적의 기능 확장을 위한 것이다. 따라서, 이하 설명되는 헤더 압축 단계에서는 상기한 4개의 필드들의 변경에 따른 조건은 고려하지 않는다.

[표 1]

필드명	프로토콜	필드 크기(octet)	식별자
Address	PPP	1	10
Control		1	11
Version	IPv4	1	12
IHL		1	13
Type of Service		1	14
Total Length NOTE#1		2	15
Flags		1	16
Fragment Offset		2	17
Time to Live NOTE#1		1	18
Protocol		1	19
Header Checksum NOTE#1		2	20
Source Address		4	21
Destination Address		4	22
Source Port	UDP	2	23
Destination Port		2	24
Frame Check Sequence NOTE#1	PPP	4	25
Version	IPv6	4	26
Traffic Class		8	27
Flow Label		20	28
Payload Length		16	29
Next Header		8	30
Hop Limit NOTE#1		8	31
Source Address		128	32
Destination Address		128	33
Options	IP	Variable	00

상기와 같이 기본 헤더 구조를 생성하면, 상기 인캡슐레이터401은 1915단계에서 상기 < 표 1> 에 있는 필드들의 값을 헤더 데이터베이스403에 저장된 이전 패킷의 값들과 비교한다. 만약, 이전 패킷의 값과 다른 필드가 검출되면, 1917단계로 진행하여 상기 도 9의 기본 압축 헤더의 M 플래그를 " 1" 로 설정한다. 그렇지 않은 경우는 기본 압축 헤더의 M 플레그를 " 0" 으로 설정한다. 여기서, 값이 변경된 필드는 기본 압축 헤더의 마지막 부분에 도 10에 나타난 형태로 추가된다. 즉, 1919단계에서 변경된 필드의 식별자를 필드 식별자(Field Identifier)에 삽입하고, 필드 정보(Fi

eld Information)에 해당 필드의 값을 삽입하여 기본 압축 헤더로 추가한다. 만약, 변경된 값을 가지는 필드가 또 다시 발견되면, 마지막에 추가된 필드 식별자(Field Identifier)의 M 플래그를 " 1" 로 설정한 후, 동일한 작업을 통하여 해당 필드를 추가한다. 목적지 포트(Destination Port) 필드까지의 점검이 마쳐지면, 마지막에 추가된 필드의 M 플래그를 " 0" 으로 설정한다.

이후, 상기 인캡슐레이터401은 1921단계에서 수신된 패킷의 IP 헤더에 옵션(Options) 필드가 있는지를 검사한다. 만약, 상기 옵션 필드가 존재하면 1923단계에서 마지막에 추가된 변경 필드의 M 플래그를 '1'로 설정하고, 이때 추가된 변경 필드가 없는 경우는 기본 압축 헤더의 M 플래그를 " 1" 로 설정한다. 그리고, 상기 인캡슐레이터401은 도 11의 구조를 갖는 부가 필드를 압축 헤더의 끝에 추가한다. 이후, 1925단계에서 필드 식별자(Field Identifier)를 '00'으로 설정하고, 추가되는 부가필드의 길이를 길이필드(LENGTH)에 추가하며, 부가필드의 내용을 추가한다. 그리고, 1927단계에서 M 플래그는 '0'으로 설정하고, 1929단계에서 상기와 같이 생성된 압축된 헤더에 사용자 데이터 필드를 추가한다. 그리고, 압축헤더를 가지는 패킷을 통신 링크를 통하여 종단으로 전송되거나 하위의 통신 프로토콜로 전달한다.

< 패킷 헤더 해제 과정>

도 20은 본 발명의 실시 예에 따른 패킷 헤더의 압축을 해제하는 과정을 도시하고 있다 압축된 패킷을 해제하는 과정은 상기한 압축을 수행하는 과정에 비해 간단하다.

상기 도 20을 참조하면, 먼저, 상기 도 9에 도시된 기본 압축 헤더의 압축 해제를 수행한다. 2001단계에서 상기 디캡슐레이터402는 해당 연결의 채널식별자(CH-ID) 필드 값을 연결식별자(CONN-ID) 값으로 복사한다. 즉, 해당 연결식별자에 해당하는 IP 소스 및 목적지 어드레스, 그리고 UDP 소스 및 목적지 포트를 압축 해제된 헤더의 값으로 저장한다. 이후, 2003단계에서 수신된 패킷의 L플래그의 값이 " 1" 로 설정되어 있는지를 검사한다. 만일, 상기 L 플래그의 값이 0이면 2025단계로 진행하여 1 옥텟의 정보를 UDP 헤더의 Length 필드 값으로 저장하고, " 1" 로 설정되어 있으면 2005단계로 진행하여 2 옥텟의 정보를 UDP 헤더의 Length 필드 값으로 저장한다.

그리고, 2007단계에서 IPv4 패킷이 수신되는지를 검사한다. 만일, IPv6 패킷이 수신되면, 2027단계로 진행하여 IP I NFO 필드를 IPV6 Next Header 정보로 설정한다. 만일, IPv4 패킷이 수신되면 2009단계로 진행하여 수신된 IP_IN FO 필드의 값을 저장되어 있는 과거의 IP_INFO값과 비교하여, 증가된 차이를 IP의 Identification 필드 값으로 저장한다. 아울러, 수신된 값을 헤더 데이터베이스의 IP_INFO필드에 저장한다. 그리고, 2011단계에서 수신된 프로토콜 필드 값을 PPP의 프로토콜 필드 값으로 저장한다. 이 경우, RFC1661의 프로토콘 헤더 압축(Protocol Header Compression) 방안을 따른다.

이후, 2013단계에서 수신된 U 플래그의 값이 " 1" 인지를 검사한다. 만일, 상기 수신된 U 플래그의 값이 1이면, 2015단계로 진행하여 수신된 Checksum 필드 값을 UDP Checksum 필드의 값으로 저장한다. 그렇지 않으면, 2009단계로 진행하여 Checksum을 계산하여 UDP 패킷 헤더로 포함시킨다.

그리고, 2017단계에서 수신된 M플래그의 값이 " 1" 인지를 검사한다. 만일, 상기 수신된 M플래그의 값이 1인 경우(다시말해, 고정된 값을 가지는 필드들이 변경된 경우) 디캡슐레이터402는 수신된 패킷으로 부터 도10의 정보를 추출한다. 상기 디캡슐레이터402는 2011단계에서 부가된 첫번째 Field Identifier를 식별한 후, 상기 < 표 1> 을 통하여 길이 정보와 해당 필드 명을 검출한다. 해당 필드의 값을 올바르게 읽은 상기 디캡슐레이터402는 2021단계에서 해당 필드의 값을 재구성되는 헤더의 필드에 저장한다. 아울러, 변경된 값들은 헤더 데이터베이스493의 해당 필드들로 복사된다. 만약, 부가 Field Identifer의 M 플래그가 0이 아닌 경우는 동일한 과정을 반복한다. 한편, Field Identifier가 "

00" 인 경우가 발생하면, 상기 디캡슐레이터402는 해당 부가 필드의 Length 필드를 읽어서 부가 필드의 길이를 검출하고, 해당 길이 만큼의 정보를 IP 헤더의 Options 필드로 복사한다. 이후, 상기 디캡슐레이터402는 2023단계에서 재구성된 패킷 헤더의 끝에 사용자 데이터 필드를 추가한다. 그리고, 압축 해제된 패킷을 상위 통신 프로토콜로 전달한다.

< 호 해제 과정>

호의 해제 과정은 단순하다. 호의 해제가 이루어지면, 상기 옵티마이저는 해당 연결에 대한 CONN-ID를 통하여 H-DB내의 관련 정보를 삭제한다. 그리고, 사용한 CONN-ID를 반납함으로서, 이후 다른 연결이 CONN-ID를 재활용할 수 있도록 한다.

본 발명에 따른 UDP/IPv4& IPv6/PPP 헤더 압축 방안을 유선 통신망과 무선 통신망에 적용할 경우의 실시예 구성을 각각 도 12와 도 13에서 도시하고 있다.

유선 통신망에서 본 발명을 적용하는 경우 도 12에 도시된 바와 같이, 2가지 환경을 고려할 수 있다. 상기 도 12a에 도시된 바와 같이, 가입자가 서버로 모뎀 장비 혹은 저속의 통신 링크를 통하여 접속하는 경우이다. 이러한 경우에 본 발명은 가입자 장비의 PPP 혹은 PPP의 하위 계층으로 적용한다. 아울러, 서버(server)의 경우에도 복수의 가입자를 지원하는 MODEM Pool 상위의 PPP의 하위 계층 혹은 PPP내의 기능으로 적용한다. 따라서, 가입자와 서버간의 UDP 서비스는 하위의 옵티마이저를 인식하지 않고 통신을 수행하며, 상기 옵티마이저는 이들간에 송수신되는 트래픽의 UDP/IP/PPP 헤더를 자동으로 압축하고 해제하므로서 유선 링크의 효율을 극대화 한다. 또한, 서버를 경유하지 않고 가입자와 가입자가 직접 연결하는 도 12b의 경우에도 가입자들이 의식하지 못하는 상태에서 자동으로 옵티마이저가 유선 링크의 효율을 극대화 하므로서, 전송률이 더욱 높아지는 결과를 가져온다. 도 12a에 도시되어 있는 클라이언트/서버 아키텍쳐(architecture) 예를 살펴보면, 클라이언트는 선로(wired line)에 연결되는 모뎀1203과 상기 모뎀1203에 연결되는 프로토콜 계층부1202(UDP/IP/PPP/Optimizer)와 상기 프로토콜 계층부1202에 연결되는 어플리케이션1201로 구성된다.한편, 서버는 선로에 연결되며 복수의 가입자들을 지원하기 위한 모뎀풀(MODEM Pool)과 상기 모뎀풀에 연결되는 프로토콜계층부(UDP/IP/PPP/Optimizer)와 어플리케이션(application)으로 구성된다.그리고, 도 12b에 도시되어 있는 종단간 아키텍쳐 예를 살펴보면, 종단에 위치하는 두 개의 종단 터미널은 동일한 구성을 가지며, 상기 종단 터미널은 선로에 연결되는 모뎀1203과 상기 모뎀1203에 연결되는 프로토콜 계층부1202(UDP/IP/PPP/Optimizer)와 상기 프로토콜 계층부1202에 연결되는 어플리케이션1201로 구성된다

무선 통신망에서 본 발명을 적용하는 경우 도 13에서와 같이, PDSN(Packet Data Serving Node) 혹은 IWF(Inter-Working Function)에 옵티마이저가 적용될 수 있다. 이는 PPP 계층의 부가 기능으로서 본 발명을 적용하는 경우이다. 그렇지 않은 경우에는 BSS(Base-Station System) 시스템의 RF(Radio Frequency) 계층(Layer)2 프로토콜에서 본 발명을 적용할 수 있다. 이는 PDSN/IWF가 헤더 압축을 인식하지 못하는 환경에서 운용된다. 무선망에 본 발명을 적용하는 경우도 유선망에서와 마찬가지로 향상된 전송 효율을 지원할 수 있으며, 특별히 인터넷 전화의 지원이 가능해진다. 즉, 기존의 IS95A, IS95B, PCS와 같기나 이하의 무선 자원을 요청하는 인터넷 전화의 지원이 가능하다. 상기 도 13에 도시되어 있는 무선망에서의 아키텍쳐 예를 살펴보면, 크게 이동단말(mobile terminal), 기지국시스템(Base Station system), PDSN/IWF로 구성되며, 각각의 구성에 대한 프로토콜 탑재 상태를 살펴보면 다음과 같다. 먼저, 이동단말은, 상기 기지국 시스템과 무선통신을 위한 RF계층1 및 RF계층2 그리고 그 상위에 UDP/IP/PPP/Optimizer 로 구성되는 프로토콜 계층부1302와 그 상위의 어플리케이션1301로 구성된다. 그리고, 상기 기지국 시스템1303은 상기 무선단말과 통신하기 위한 RF계층1 및 RF계층2와 상기 PDSN/IWF와 통신하기 위한 계층1 및 계층2와 상위 프로세서(Mobile processor)로 구성된다. 한편, 상기 PDSN/IWF는 상기 기지국 시스템과 통신하기 위한 계층1 및 계층2와 그 상위에 optimizer, PPP,IP, UDP 로 구성되는 프로토콜 계층부1305와 그 상위의 PDSN/IWF 모듈(module)1304로 구성된다.

도 14는 CDMA2000에서 RLP Layer2의 RLP 프레임 구조를 도시하고 있다. 기본채널(FCH:fundamental channel)과 전용제어채널(DCCH:dedicated control channel)에서 RLP의 투명한 모드(transparent mode)를 적용하는 경우 지원할 수 있는 트래픽 용량은 각각 160bits/20ms와 256bits/20ms 이다. 앞서 설명한 바와 같이 G.729와 G.723.1 프로토콜은 UDP/IP/PPP의 헤더가 최소 6바이트로 압축된 환경에서 하기 표 2에 나타난 것과 같이 성공적으로 지원될 수 있다. 아울러, 헤더의 크기가 20바이트나 증가한 IPv6에 대해서도 하기 표 3과 같이 성공적으로 지원할 수 있다.

[표 2]

	DCCH/FCH - 9.6kbps(160bits/20ms)	DCCH/FCH - 14.4kbps(256bits/20ms)
G.729 (688bits/20ms)	지원 불가능	지원 불가능
G.729 + 헤더 압축 방안 (256bits/20ms)	지원 불가능	지원 가능
G.723.1 6.4 (456bits/30ms)	지원 불가능	지원 불가능
G.723.1 6.4 kbps + 헤더 압축 방안 (240bits/30ms)	지원 가능	지원 가능
G.723.1 5.3 (424bits/30ms)	지원 불가능	지원 불가능
G.723.1 5.3 kbps + 헤더 압축 방안 (208bits/30ms)	지원 가능	지원 가능

[표 3]

	DCCH/FCH - 9.6kbps(160bits/20ms)	DCCH/FCH - 14.4kbps(256bits/20ms)
G.729 (848bits/20ms)	지원 불가능	지원 불가능
G.729 + 헤더 압축 방안 (256bits/20ms)	지원 불가능	지원 가능
G.723.1 6.4 (616bits/30ms)	지원 불가능	지원 불가능
G.723.1 6.4 kbps + 헤더 압축 방안 (240bits/30ms)	지원 가능	지원 가능
G.723.1 5.3 (584bits/30ms)	지원 불가능	지원 불가능
G.723.1 5.3 kbps + 헤더 압축 방안 (208bits/30ms)	지원 가능	지원 가능

상기 표 2 및 표 3에서와 같이, G.729와 H.723.1의 헤더를 압축하지 않으면, DCCH(Dedicated Control Channel)/FCH(Fundamental Channel) 상에서 지원이 불가능했다. 그러나, 본 발명의 UDP/IP/PPP 헤더 압축 기법을 사용함으로서 G.729와 G.723.1은 무선 환경에서도 기존 회선형 전화와 동일하거나 낮은 비용으로 지원이 가능하다.

이상은 UDP/IPv4& IPv6/PPP 프로토콜의 패킷 헤더를 압축하는 방안에 대하여 기술하였다. 본 발명에서는 UDP/IPv4& IPv6/PPP 패킷의 헤더를 압축하는 방안과 함께 실시간 멀티미디어 서비스의 지원에 많이 활용되고 있는 RTP 프로토콜의 헤더를 압축하는 방안을 함께 제안한다. 상기 RTP의 헤더는 압축을 거치지 않고 앞서 제안한 UDP/IPv4& IPv6/PPP 헤더 압축 방안에 따른 헤더의 트래픽으로서 전달될 수 있다. 그러나, 본 발명에서는 지속의 유무선 환경에서 RTP의 헤더 부분도 압축하므로 보다 높은 효율을 제공하는 방안을 제안한다.

상기 RTP 프로토콜의 헤더 구조가 도 15에 도시되어 있다. RTP의 프로토콜 헤더 중에서 V, X, CC, M, PT, SSRC 및 CSRC 필드는 초기 호 설정이 있은 후에는 거의 변경되지 않는 필드들이다. 따라서, 패킷의 전송시에 변경되는 필드들은 시퀀스번호(Sequence Number)와 타임스탬프(Timestamp)로 국한된다. 따라서, 본 발명에서는 앞서 제안한 간략화된 UDP/IPv4& IPv6/PPP 헤더와 RTP 헤더를 통합하므로서, 압축 효과를 극대화하는 접근에 대해 설명한다.

본 발명에 따른 RTP/UDP/IPv4& IPv6/PPP 프로토콜 패킷의 통합 압축에 따른 헤더 구조가 도 16와 도 17에 도시되어 있다. 상기한 두 헤더 구조의 차이점은 UDP CHECKSUM 필드가 포함되거나 포함되지 않는 구조로서, 만약 트래픽 속성이 에러에 민감하지 않거나 크게 에러에 영향을 받지 않는 트래픽인 경우는 UDP CHECKSUM 필드를 제거하고, 그렇지 않은 경우는 UDP CHECKSUM 필드를 포함한다. 이 경우, UDP CHECKSUM 필드를 포함하지 않는 경우는 U 플래그를 "0" 으로 설정하고, UDP CHECKSUM 필드가 포함되는 경우는 1로 설정된다.

상기한 UDP/IPv4& IPv6/PPP의 기본 압축 헤더에 추가되는 필드들을 설명하면 다음과 같다.

먼저, RTP 헤더가 압축되는 경우에는 R 플래그를 1로 설정하여, 수신단 디캡슐레이터402가 해당 패킷내에 RTP 헤더 정보가 포함되었음을 인식할 수 있도록 한다. 상기 RTP 헤더의 시퀀스번호(Sequence Number)를 포함시키는 방법은 다음과 같다. 상기 RTP의 지원을 위하여 상기 IP info 필드는 6비트의 sequence 필드와 2비트의 RVD 필드로 활용한다. 또한 이는 IPv4와 IPv6에 대하여 동일하게 적용한다. 상기 IPv4를 이용하는 경우에 있어서는 sequence필드가 앞서 기술한 바와 같이 상기 수학식 1에 따른 값을 가지도록 한다. 이 경우, 상기 RTP의 sequence Number 증가 수준이 IPv4의 패킷 번호 증가치와 동일하면, 헤더의 RVD 필드는 '00'의 값을 가진다. 상기 RTP 헤더와 압축된 패킷의 헤더를 수신한 디캡슐레이터402는 상기 RTP의 sequence 값을 IPv4 identification의 증가치와 동일한 수준에서 증가시킨다. 만약, RTP의 sequence number가 IPv4의 증가치보다 '1'이 많다면, RVD는 '01'로 설정되며, 이를 수신한 디캡슐레이터402는 재구성되는 RTP 헤더의 sequence number를 IPv4 identification의 증가치보다 '1' 많게 증가시킨다. 역으로, RTP의 sequence number가 '10'으로 설정되어 있는 경우는 RTP의 sequence number가 IP의 sequence number의 증가치보다 1 작은 경우로서, 수신단의 디캡슐레이터402는 IPv4 identification 증가치보다 1 작은 증가치로 RTP 헤더를 재구성한다. 만약, RTP sequence number의 증가치가 2 이상인 경우는 RVD필드를 11로 설정하고, 증가된 값을 나타내는 1 octet을 UDP필드 뒤의 UDP/IPv4/PPP가 부가적으로 추가한 옵션 필드(optional field)의 마지막에 추가한다.

IPv6를 사용하는 경우에는 하기 수학식 2에 따라 IP info 필드를 채운다.

수학식 2

$$\text{IP info} = \text{RTP sequence number} \ \% \ 65$$

그리고, Simple Time-Stamp 필드는 하기 < 수학식 3> 에 의해 기존의 Time-stamp를 대치한다.

수학식 3

$$\text{Simple Time-Stamp} = \text{RTP Time-Stamp} \ \% \ 4096$$

한편, 수신단 디캡슐레이터402는 저장된 Time-Stamp 값에 Simple-Time-Stamp값의 증가치를 더하여 RTP 헤더를 재구성한다. 따라서, 상기 인캡슐레이터401과 상기 디캡슐레이터402는 초 설정시에 Simple Time-Stamp 필드를 생성하여 보관해야 한다. 상기 RTP 헤더의 V, X, CC 필드 값이 변경되는 경우에 상기 인캡슐레이터401은 H 플래그를 "1" 로 설정하여 전송한다. 상기 H 플래그가 1로 설정되면, 헤더의 마지막에 RTP의 처음 1바이트 헤더 정보(V, P, X, CC 필드)를 추가한다. 이 경우, X 필드의 값이 변경된 경우에는 RTP의 X 플래그에 연계된 RTP의 부가 정보 필드들이

추가로 포함된다. CC 필드가 변경된 경우에는 RTP의 CSRC 필드의 정보들이 추가된다.

RTP의 PT 필드의 정보가 변경되는 경우 P 플래그를 1로 설정하며 그렇지 않은 경우는 0으로 설정한다. 이때 상기 P 플래그가 설정된 경우에는 헤더의 마지막에 PT의 정보 전송을 위한 1 octet의 필드가 추가되고, 해당 필드에 PT 정보를 저장한다. RTP의 SSRC 필드 값이 변경된 경우 S 플래그를 "1"로 설정하며, 그렇지 않은 경우에는 0으로 설정한다. 상기 S 플래그가 설정되면, 헤더의 마지막에 SSRC 정보 전송을 위한 4 octet 필드가 추가되고, 해당 필드에 SSRC의 정보가 저장된다. Q 플래그는 이후를 위하여 예약된 비트이다. 이러한 과정을 수행하므로서 RTP 헤더는 2바이트의 추가 필드로서 앞서 제안한 UDP/IP/PPP 헤더의 기본 구조에 추가된다.

발명의 효과

상술한 바와 같이, 본 발명을 통하여 최소한 36 바이트를 요구하는 UDP/IP/PPP 프로토콜의 헤더 구조가 4~6 바이트 수준으로 축소된다. 따라서, 저속의 유무선 환경에서 작은 통신 용량을 되도록이면 사용자 트래픽의 송수신에 사용할 수 있도록 함으로서 통신 링크의 효율을 증가시킬수 있다. 이는 사용자 입장에서 통신 링크의 고속화를 제공할 수 있다. 그리고 본 발명을 통하여 비연결형 UDP 통신 프로토콜에 기반하는 실시간 멀티미디어 서비스들의 지원이 가능해진다. 특히, IS95A, IS95B, PCS 및 IMT2000에서도 기존의 회선형 전화가 무선에서 9.6 혹은 14.4kbps를 지원하는데, 이와 동일하거나 그보다 낮은 용량에서 필요로 하는 인터넷 전화 서비스를 무선 환경에서 제공할 수 있다. 이는 기존의 회선형 무선 전화가 'MS-BSS-MSC(Mobile Switching Center)'의 경로를 가지면서 유선망에서 고가의 전용회선을 사용해야하던 문제점을 해결하여, 'MS-BSS-PDSN/IWF'의 패킷 유선망을 지원할 수 있도록 한다. 따라서, 유선망에 요구되는 비용이 현격히 줄어드는 결과를 야기한다. 또한,본 발명은 추가적으로 RTP 프로토콜을 UDP/IP/PPP 프로토콜의 상위에서 지원하는 방안을 제공한다. 이 경우 52바이트를 요구하는 RTP/UDP/IP/PPP 프로토콜 헤더가 6~8 바이트 수준으로 축소된다. 이를 통하여, RTP를 이용하는 기존의 유선 인터넷 전화 단말 혹은 교환기들과도 투명성있게 인터넷 전화를 지원할 수 있다.

(57) 청구의 범위

청구항 1.

통신시스템에서 UDP/IP/PPP 헤더 압축 장치에 있어서,

상기 프로토콜 계층으로부터 UDP, IP, PPP 패킷들을 수신하며, 상기 패킷들에서 헤더를 추출하여 미리 정해진 크기의 압축헤더로 압축하여 하위 프로토콜 계층으로 전달하는 인캡슐레이터와,

통신링크를 통해 압축된 패킷을 수신하며, 상기 압축된 패킷을 해제하여 원래의 UDP, IP, PPP 패킷들로 분해하여 상기 상위 프로토콜 계층으로 전달하는 디캡슐레이터와,

상기 인캡슐레이터와 상기 디캡슐레이터의 헤더 압축 및 해제를 위한 정보들을 저장하는 헤더 데이터베이스를 포함하는 것을 특징으로 하는 장치.

청구항 2.

제1항에 있어서,

상기 PPP의 연결설정 및 연결해제 관련 기능을 구동하는 연결추적기(connection tracer)를 더 구비함을 특징으로 하는 장치.

청구항 3.

제1항에 있어서, 상기 압축헤더는,

설정된 통신링크에 대한 연결식별정보를 저장하는 필드(CH-ID)와,

상기 압축 헤더의 길이정보를 저장하는 필드(LENGTH)와,

상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 저장하는 필드(IP-INFO)와,

상기 PPP 패킷의 프로토콜필드(PROTOCOL) 값을 저장하는 필드(PROTOCOL)와,

상기 UDP 패킷의 체크섬필드(CHECKSUM) 값을 저장하는 필드(UDP CHECKSUM)와,

변경된 필드 및 부가적인 필드 값들을 저장하기 위한 필드(OPTIONAL FIELD)를 포함하는 것을 특징으로 하는 장치.

청구항 4.

제1항에 있어서,

상기 IP 패킷은 IPv4와 IPv6 중 어느 하나인 것을 특징으로 하는 장치.

청구항 5.

제1항에 있어서, 상기 헤더 데이터베이스는,

설정된 통신링크를 식별하기 위한 연결식별정보(CONN-ID)와,

상기 IP 패킷을 구분하기 위한 정보(IP TYPE)와,

상기 UDP, IP, PPP 패킷들의 헤더들에 저장되어 있는 필드값들을 저장하는 것을 특징으로 하는 장치.

청구항 6.

UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 UDP,IP,PPP 패킷 헤더들을 소정 크기의 헤더로 압축한 압축 헤더 구조에 있어서,

설정된 통신링크에 대한 연결식별정보를 저장하는 필드(CH-ID)와,

상기 UDP 패킷의 길이정보를 저장하는 필드(LENGTH)와,

상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 저장하는 필드(IP-INFO)와,

상기 PPP 패킷의 프로토콜필드(PROTOCOL) 값을 저장하는 필드(PROTOCOL)와,

상기 UDP 패킷의 체크섬필드(CHECKSUM) 값을 저장하는 필드(UDP CHECKSUM)를 포함하는 것을 특징으로 하는 압축 헤더구조.

청구항 7.

제6항에 있어서, 상기 압축 헤더구조는,

이전 패킷과 비교하여 변경된 필드에 대한 식별정보를 저장하는 필드(FIELD IDENTIFIER)와,

상기 변경된 필드의 필드값을 저장하는 필드(FIELD INFORMATION)를 더 포함하는 것을 특징으로 하는 압축 헤더구조.

청구항 8.

제6항에 있어서, 상기 압축 헤더구조는,

상기 IP 패킷의 부가필드(OPTION) 존재여부를 나타내는 필드와,

상기 부가필드의 길이 정보를 저장하는 필드와,

상기 부가필드 값을 저장하는 필드를 더 포함하는 것을 특징으로 하는 압축 헤더구조.

청구항 9.

RTP/UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 RTP, UDP,IP,PPP 패킷 헤더들을 소정 크기의 헤더로 압축한 압축 헤더 구조에 있어서,

설정된 통신링크에 대한 연결식별정보를 저장하는 필드(CH-ID)와,

상기 UDP 패킷의 길이정보를 저장하는 필드(LENGTH)와,

상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 저장하는 필드(SEQUENCE)와,

상기 RTP 패킷의 시퀀스번호(SEQUENCE NUMBER) 증가분과 상기 IP 패킷의 패킷번호 증가분과의 관계를 나타내는 정보를 저장하는 필드(RVD)와,

상기 RTP 패킷의 타임스탬프(TIME STAMP)를 소정 규칙에 의해 수정하여 저장하는 필드(SIMPLE TIME STAMP)를 포함하는 것을 특징으로 하는 압축 헤더 구조.

청구항 10.

제9항에 있어서, 상기 압축 헤더 구조는,

상기 RTP 패킷의 필드 값들의 변경여부를 나타내는 식별필드(H.P.S.Q)와,

상기 변경된 필드들의 값들을 저장하는 필드(OPTIONAL FIELD)를 더 포함하는 것을 특징으로 하는 압축 헤더구조.

청구항 11.

제9항에 있어서, 상기 압축 헤더 구조는,

상기 UDP 패킷의 체크섬필드(CHECKSUM) 값을 저장하는 필드(UDP CHECKSUM)를 더 포함하는 것을 특징으로 하는 압축 헤더구조.

청구항 12.

UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 UDP,IP,PPP 패킷 헤더들을 소정 크기의 압축헤더로 압축하기 위한 방법에 있어서,

상기 압축헤더는 L,R,U,M 플래그, 채널식별자 필드(CH-ID), 길이정보 필드(LENGTH), IP정보필드(IP-INFO), 프로토콜 필드(PROTOCOL), 체크섬 필드(UDP CHECKSUM), 부가필드(OPTIONAL FIELD)를 적어도 포함하며,

상위 프로토콜 계층부로부터 상기 UDP,IP,PPP 패킷들을 수신하고, 설정된 통신링크에 대한 연결식별정보를 상기 채널식별자 필드에 기록하고 상기 R플래그를 세팅하는 과정과,

상기 UDP 패킷의 길이 필드 값을 상기 길이정보 필드에 기록하고, 상기 L플래그를 설정하는 과정과,

상기 IP 패킷의 식별필드(IDENTIFICATION) 값을 소정 규칙에 의해 수정하여 상기 IP정보필드에 기록하는 과정과,

상기 PPP 패킷의 프로토콜 필드 값을 상기 프로토콜 필드에 기록하는 과정과,

상기 UDP 패킷의 체크섬 필드 값을 상기 체크섬 필드에 기록하고, 상기 U플래그를 세팅하여 기본압축헤더를 생성하는 과정을 포함함을 특징으로 하는 방법.

청구항 13.

제12항에 있어서,

상기 채널식별자 필드는 4비트로 구성되는 것을 특징으로 하는 방법.

청구항 14.

제12항에 있어서,

상기 PPP 패킷의 프로토콜 필드 값은 RFC1661의 프로토콜 헤더 압축에 따라 8비트로 고정되어 상기 프로토콜 필드에 기록되어지는 것을 특징으로 하는 방법.

청구항 15.

제12항에 있어서,

상기 UDP,IP,PPP 패킷들을 이전 패킷들과 비교하여 변경된 필드가 존재하는 경우, 상기 M플래그를 세팅하고, 상기 변경된 필드를 식별하기 위한 식별정보를 기록하는 필드와 상기 변경된 필드의 필드값을 기록하는 필드를 상기 부가필드에 복사하는 과정을 더 포함하는 것을 특징으로 하는 방법.

청구항 16.

제12항에 있어서,

상기 IP 패킷에 부가필드(OPTION)가 존재하는 경우, 상기 M플래그를 세팅하고, 상기 IP 패킷의 부가필드임을 나타내는 식별필드와 상기 부가필드의 길이정보를 기록하는 필드와 상기 부가필드의 필드값을 기록하는 필드를 상기 부가필드에 복사하는 과정을 더 포함하는 것을 특징으로 하는 방법.

청구항 17.

제12항에 있어서,

상기 IP 패킷의 식별필드 값은 하기 수학식4에 의해 수정되어짐을 특징으로 하는 방법.

수학식 4
IP정보필드 값 = IPv4 식별필드 값 % 65

청구항 18.

UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 UDP,IP,PPP 패킷 헤더들을 소정크기로 압축한 압축헤더를 압축 해제하기 위한 방법에 있어서,

상기 압축헤더는 L,R,U,M 플래그, 채널식별자 필드(CH-ID), 길이정보 필드(LENGTH), IP정보필드(IP-INFO), 프로토콜 필드(PROTOCOL), 체크섬 필드(UDP CHECKSUM), 부가필드(OPTIONAL FIELD)를 적어도 포함하며,

상기 압축헤더를 가지는 패킷수신시 상기 채널식별자 필드에 대응하는 연결식별자를 검출하고, 상기 연결식별자에 대응하는 정보들을 압축 해제된 헤더들의 해당 필드들에 기록하는 과정과,

상기 L플래그의 세팅여부에 따라 서로 다른 길이의 길이정보를 상기 UDP 패킷의 길이필드에 기록하는 과정과,

상기 IP정보필드와 이전 패킷의 IP정보필드 값을 비교하여 증가분을 상기 IP패킷의 식별필드(identification)에 기록하는 과정과,

상기 프로토콜 필드 값을 상기 PPP 패킷의 프로토콜 필드에 기록하는 과정과,

상기 U플래그가 세팅되어 있을 시 상기 체크섬필드 값을 상기 UDP 패킷의 체크섬필드에 기록하는 과정과,

상기 M플래그가 세팅되어 있을 시 상기 부가필드에 기록되어 있는 필드들을 식별하여 상기 압축 해제된 헤더들의 해당 필드들로 복사하는 과정을 포함하는 것을 특징으로 하는 방법.

청구항 19.

RTP/UDP/IP/PPP 통신 프로토콜에 기반하는 통신시스템에서 RTP,UDP,IP,PPP 패킷 헤더들을 소정 크기의 압축헤더로 압축하기 위한 방법에 있어서,

상기 압축헤더는 L,R,U,M,H,P,S,Q 플래그, 채널식별자 필드(CH-ID), 길이정보 필드(LENGTH), IP정보필드(IP-INFO), 프로토콜 필드(PROTOCOL), 심플 타임스탬프 필드(SIMPLE TIME STAMP), 부가필드(OPTIONAL FIELD)를 적어도 포함하며,

상위 프로토콜 계층부로부터 상기 RTP, UDP,IP,PPP 패킷들을 수신하고, 설정된 통신링크에 대한 연결식별정보를 상기 채널식별자 필드에 기록하고 상기 R플래그를 세팅하는 과정과,

상기 UDP 패킷의 길이 필드 값을 상기 길이정보 필드에 기록하고, 상기 L플래그를 설정하는 과정과,

상기 프로토콜 필드 값을 상기 PPP패킷의 프로토콜 필드에 기록하는 과정과,

상기 RTP 패킷 헤더가 상기 압축헤더에 추가되는 경우 상기 R플래그를 세팅하는 과정과,

상기 IP정보필드를 시퀀스필드와 RVD필드로 구분하고, 상기 RTP 패킷의 시퀀스번호 필드(SEQUENCE NUMBER) 값을 수정하여 상기 시퀀스필드에 기록하고, 상기 RTP 패킷의 시퀀스번호(SEQUENCE NUMBER) 증가분과 상기 IP 패킷의 패킷번호 증가분과의 관계를 나타내는 정보를 상기 RVD필드에 기록하는 과정과,

상기 RTP 패킷의 타임스탬프 필드 값을 수정하여 상기 심플 타임스탬프 필드에 기록하는 과정과,

상기 RTP 패킷의 V,X,CC 필드값이 변경되는 경우 상기 H플래그를 세팅하고, 변경된 필드 값을 상기 부가필드에 기록하는 과정과,

상기 RTP 패킷의 RT필드 값이 변경되는 경우 상기 P플래그를 세팅하고, 상기 변경된 RT 필드 값을 상기 부가필드에 기록하는 과정과,

상기 RTP 패킷의 SSRC 필드 값이 변경되는 경우 상기 S플래그를 세팅하고, 상기 변경된 SSRC 필드 값을 상기 부가필드에 기록하는 과정을 포함하는 것을 특징으로 하는 방법.

청구항 20.

제19항에 있어서,

상기 IP 패킷은 IPv4 및 IPv6 패킷중 어느 하나인 것을 특징으로 하는 방법.

청구항 21.

제19항에 있어서,

상기 RTP 패킷의 시퀀스번호 필드 값을 하기 수학식 5에 의해 수정하여 상기 시퀀스필드에 기록하는 것을 특징으로 하는 방법.

수학식 5

시퀀스필드 값 = RTP 시퀀스번호 필드 값 % 65

청구항 22.

제19항에 있어서,

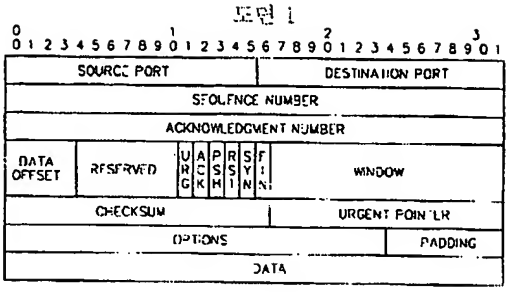
상기 RTP 패킷의 타임스탬프 필드 값을 하기 수학식 6에 의해 수정하여 상기 심플 타임스탬프 필드에 기록하는 것을 특징으로 하는 방법.

수학식 6

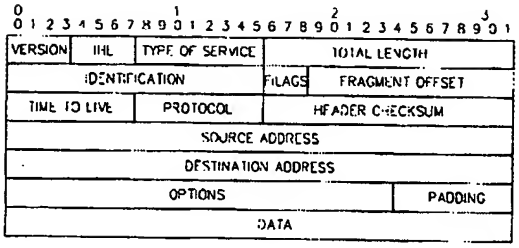
심플 타임스탬프 필드 값 = RTP 타임스탬프 필드 값 % 4096

도면

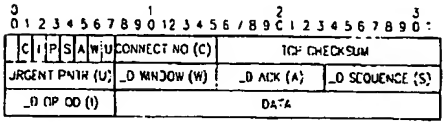
도면 1



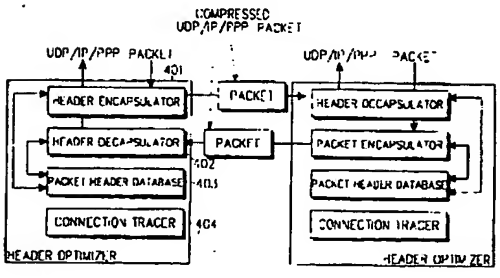
도면 2



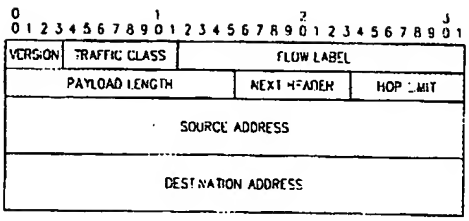
도면 3



도면 4



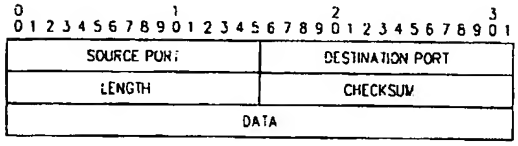
도면 5



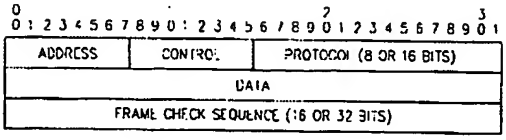
도면 6

CONN-ID	IP TYPE	FIELD #1 SOURCE PORT	FIELD #2 DESTIN PORT	...	FIELD #N FCS
CID-1	IPv4	90	80	...	0023
CID-2	IPv6	70	70	...	4598
				...	
CID-M	IPv4	66	66	...	3561

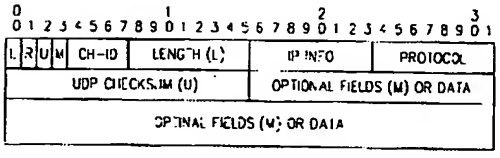
도면 7



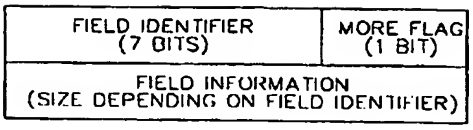
도면 8



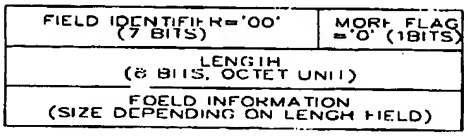
도면 9



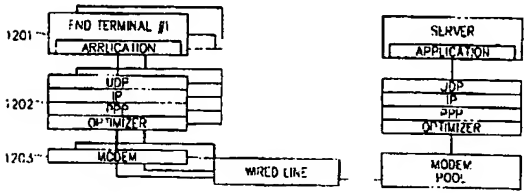
도면 10



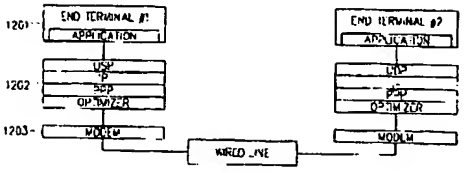
도면 11



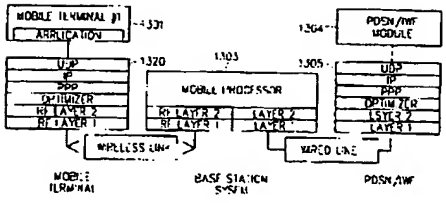
도면 12a



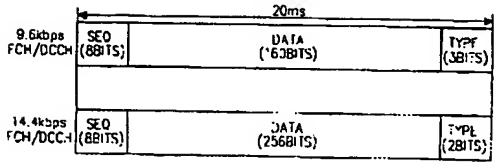
도면 12b



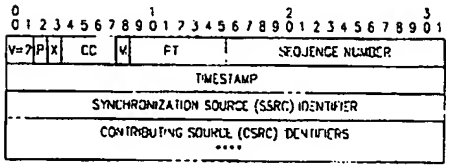
도면 13



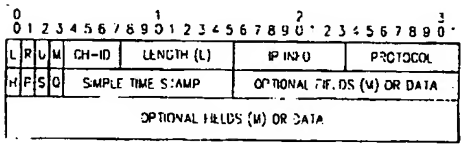
도면 14



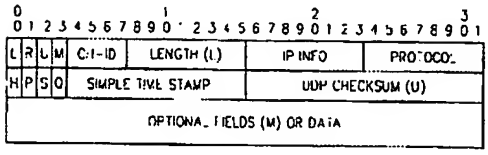
도면 15



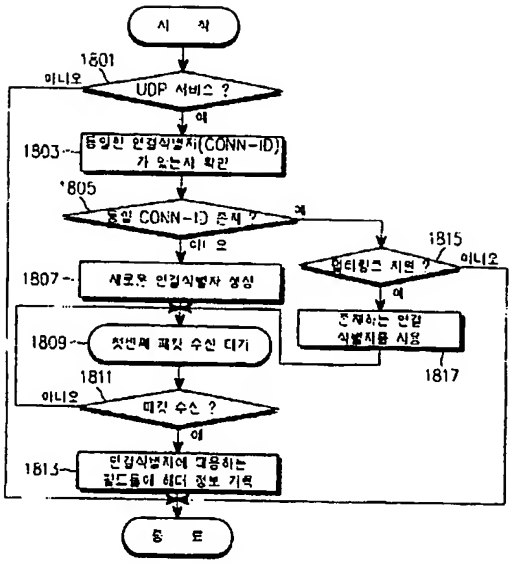
도면 16



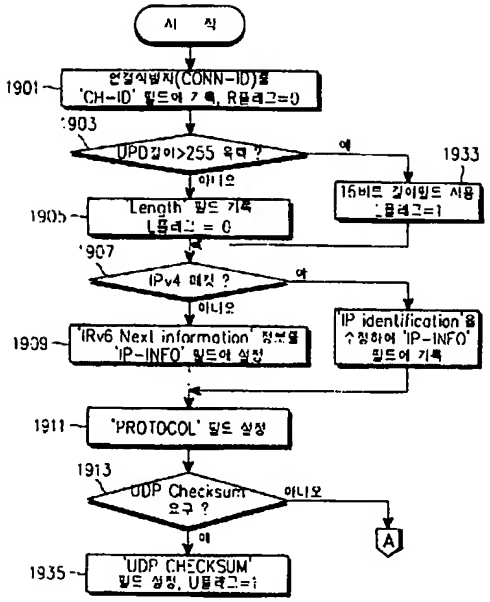
도면 17



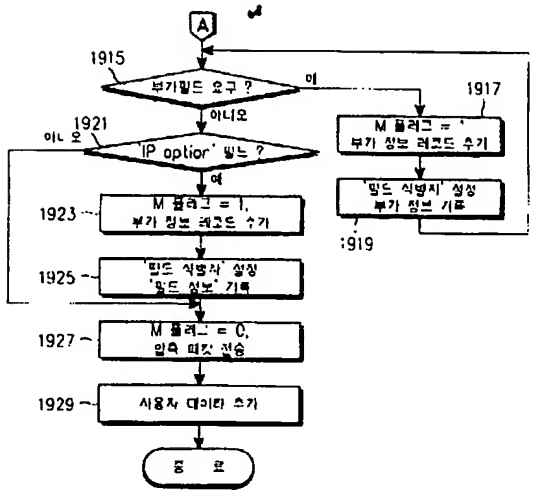
도면 18


시 작
1801
아니오
UDP 서비스 ?
예
1803
동일한 연결식별자(CONN-ID)
가 있는지 확인
1805
동일 CONN-ID 존재 ?
예
아 오
1815
멀티캐스 지원 ?
아니오
예
1807
새로운 연결식별자 생성
존재하는 연결
식별자를 사용
1817
1809
첫번째 패킷 수신 대기
1811
아니오
마지 수신 ?
예
1813
연결식별자에 대응하는
리스트에 헤더 정보 기록
종 료

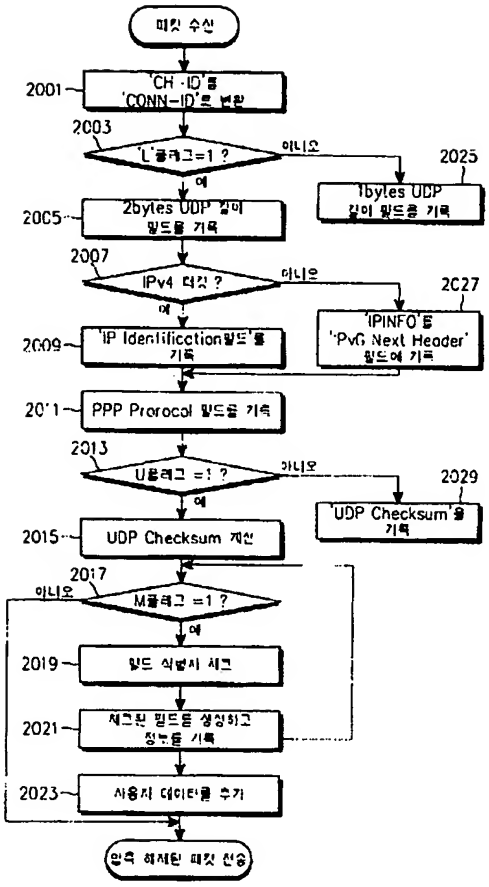
도면 19a


시 작
1901
연결식별자(CONN-ID)를
'CH-ID' 필드에 기록, R플래그=0
1903
UPD길이>255 옥텟 ?
예
1933
16비트 길이필드 사용
플래그=1
아니오
1905
'Length' 필드 기록
L플래그 = 0
1907
IPv4 헤딩 ?
예
아니오
1909
'IRv6 Next information' 정보를
'IP-INFO' 필드에 설정
'IP identification'을
수정하여 'IP-INFO'
필드에 기록
1911
'PROTOCOL' 필드 설정
1913
UDP Checksum
요구 ?
아니오
예
A
1935
'UDP CHECKSUM'
필드 설정, U플래그=1

도면 19b


A
1915
부가필드 요구 ?
예
1917
M 플래그 =
부가 정보 레코드 수기
아니오
1921
아니오
'IP option' 필드 ?
예
'필드 식별자' 설정
부가 정보 기록
1919
1923
M 플래그 = 1,
부가 정보 레코드 수기
1925
'필드 식별자' 설정
'필드 정보' 기록
1927
M 플래그 = 0,
입력 데이터 전송
1929
사용자 데이터 추가
종 료

도면 20


패킷 수신
2001
'CH -ID'를 'CONN-ID'로 변환
2003
'L'플래그=1 ?
아니오
예
2025
1bytes UDP 길이 필드를 기록
2005
2bytes UDP 길이 필드를 기록
2007
IPv4 여부 ?
아니오
예
2027
'IPINFO'를 'PvG Next Header' 필드에 기록
2009
'IP Identificction필드'를 기록
2011
PPP Prorocol 필드를 기록
2013
U플래그 =1 ?
아니오
예
2029
'UDP Checksum'을 기록
2015
UDP Checksum 계산
2017
M플래그 =1 ?
아니오
예
2019
필드 식별자 세그
2021
세그먼 필드를 생성하고 정보를 기록
2023
사용자 데이터를 추가
인접 해제된 패킷 전송